

TD 5 - Problèmes

1. Cartes de crédit

L'algorithme de Luhn fut développé dans les années 1960 pour la validation de nombres (entiers). C'est une simple formule de vérification de somme (Checksum) utilisée pour valider une variété de numéros de comptes, comme les numéros de cartes de crédit. Le principe est le suivant :

Tout d'abord, on modifie le nombre de la manière suivante : on multiplie par deux un chiffre sur deux, en commençant par l'avant dernier et en se déplaçant de droite à gauche. Si un chiffre qui est multiplié par deux donne un nombre plus grand que 9 (comme c'est le cas par exemple pour 8 qui devient 16), alors il faut le ramener à un chiffre entre 1 et 9 en additionnant les chiffres qui le composent (pour le chiffre 8 : on obtient d'abord 16 en le multipliant par 2, puis 7 en sommant les chiffres composant le résultat : 1+6). La somme de tous ces chiffres est ensuite effectuée. Enfin, le résultat est divisé par 10. Si le reste de la division est égal à zéro, alors le nombre original est valide.

Exemple

Considérons l'identification du nombre 972 487 086. Les cases colorées dans le tableau suivant indiquent les chiffres doublés. La somme, égale à 50, est divisée par 10 : le reste est 0, donc le nombre est valide.

Nombre initial	9	7	2	4	8	7	0	8	6
Doublément d'un chiffre sur 2	9	14	2	8	8	14	0	16	6
Si le double comprend 2 chiffres, on ajoute ces 2 chiffres	9	5	2	8	8	5	0	7	6
Somme des chiffres	Somme = 50								

Écrire un programme qui demande à l'utilisateur de rentrer le numéro de sa carte de crédit (sous forme d'entier) et qui vérifie si celui-ci est valide selon l'algorithme de Luhn. D'autre part le programme devra indiquer si la carte correspond à une carte Visa, une MasterCard ou un autre type. Le numéro d'une carte Visa commence toujours par le chiffre 4 et comporte entre 13 et 16 chiffres. Le numéro d'une Master Card commence par 54 ou 55 et comporte 16 chiffres.

2. Suite de Fibonacci

Écrire un programme qui affiche tous les termes d'une suite de Fibonacci. On affichera au maximum 20 termes, ceux-ci devant être inférieurs à 1000.

Définitions :

$$U_{n+1} = U_n + U_{n-1} \quad U_0 \text{ et } U_1 \quad \text{entiers positifs entrés par l'utilisateur.}$$

Affichage :

$$\begin{aligned} U_0 &= 5 \\ U_1 &= 3 \\ U_2 &= 8 \\ U_3 &= 11 \dots \end{aligned}$$

